

## BAB III

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

Kegiatan kerja magang ini dilaksanakan pada bagian teknologi Pajak.io, yaitu bagian yang mengembangkan aplikasi *web* Pajak.io. Bagian ini dipimpin oleh Jefriansyah Hertikawan selaku CTO (*Chief Technology Officer*) Fintax. Kegiatan magang ini mengerjakan proyek *User Management* yang dipimpin oleh Fazar Syukri Nugroho selaku *Backend Engineer, Engineering Lead*, dan sekaligus *supervisor* magang.

#### 3.2 Tugas dan Uraian Kerja Magang

##### 3.2.1 Tugas yang Dilakukan

Kegiatan kerja magang ini befokus pada pengembangan fungsi-fungsi dan fitur-fitur baru pada aplikasi pajak.io. Aplikasi pajak.io terbagi menjadi beberapa bagian. Pada kesempatan kegiatan kerja magang ini, yang dikembangkan adalah *User Management*. Bagian ini mengurus pendaftaran pengguna dan perusahaan baru serta pembayaran langganan Pajak.io. Kegiatan kerja magang ini akan mengerjakan fitur pembayaran langganan otomatis yang sebelumnya belum dibangun.

Tabel 3.1 merupakan timeline kerja magang pada Fintax dari tanggal 19 Mei 2021 sampai tanggal 17 November 2021 sebanyak 125 hari.

Tabel 3.1 Timeline Kerja Magang

| No | Pekerjaan yang dilakukan   | Minggu  | Tanggal Mulai    | Tanggal Selesai  |
|----|--|---------|------------------|------------------|
| 1  | Pengenalan perusahaan dan pengenalan standar <i>software engineering</i> di perusahaan | 1       | 19 Mei 2021      | 21 Mei 2021      |
| 2  | Mengimplementasikan antarmuka daftar paket langganan                                   | 2 - 7   | 24 Mei 2021      | 2 Juli 2021      |
| 3  | Mengimplementasikan antarmuka pembayaran   | 8 - 13  | 5 Juli 2021      | 13 Agustus 2021  |
| 4  | Mengimplementasikan antarmuka status pembayaran  | 14 - 20 | 16 Agustus 2021  | 1 Oktober 2021   |
| 5  | Mengimplementasikan antarmuka status penggunaan paket                                  | 21 - 23 | 4 Oktober 2021   | 22 Oktober 2021  |
| 6  | Mengimplementasikan email – email yang berkaitan dengan langganan                      | 24 - 26 | 25 Oktober 2021  | 12 November 2021 |
| 7  | Presentasi akhir magang dan offboarding  | 27      | 15 November 2021 | 17 November 2021 |

## **3.2.2 Uraian Kerja Magang**

### **3.2.2.1 Pengenalan perusahaan dan pengenalan standar software engineering di perusahaan (Minggu 1)**

Pada minggu pertama magang, HR mengenalkan perusahaan Fintax dan aplikasi Pajak.io dengan melakukan presentasi. Kemudian, HR memberikan tur kantor daring di platform *Gather*. Setelah itu, diberikan akses akun Slack (*chat*), Google Workspace (email dan layanan Google), dan GitLab (*code repository*). Hal ini dilakukan untuk mempersiapkan segala kebutuhan administrasi untuk mengerjakan tugas magang.

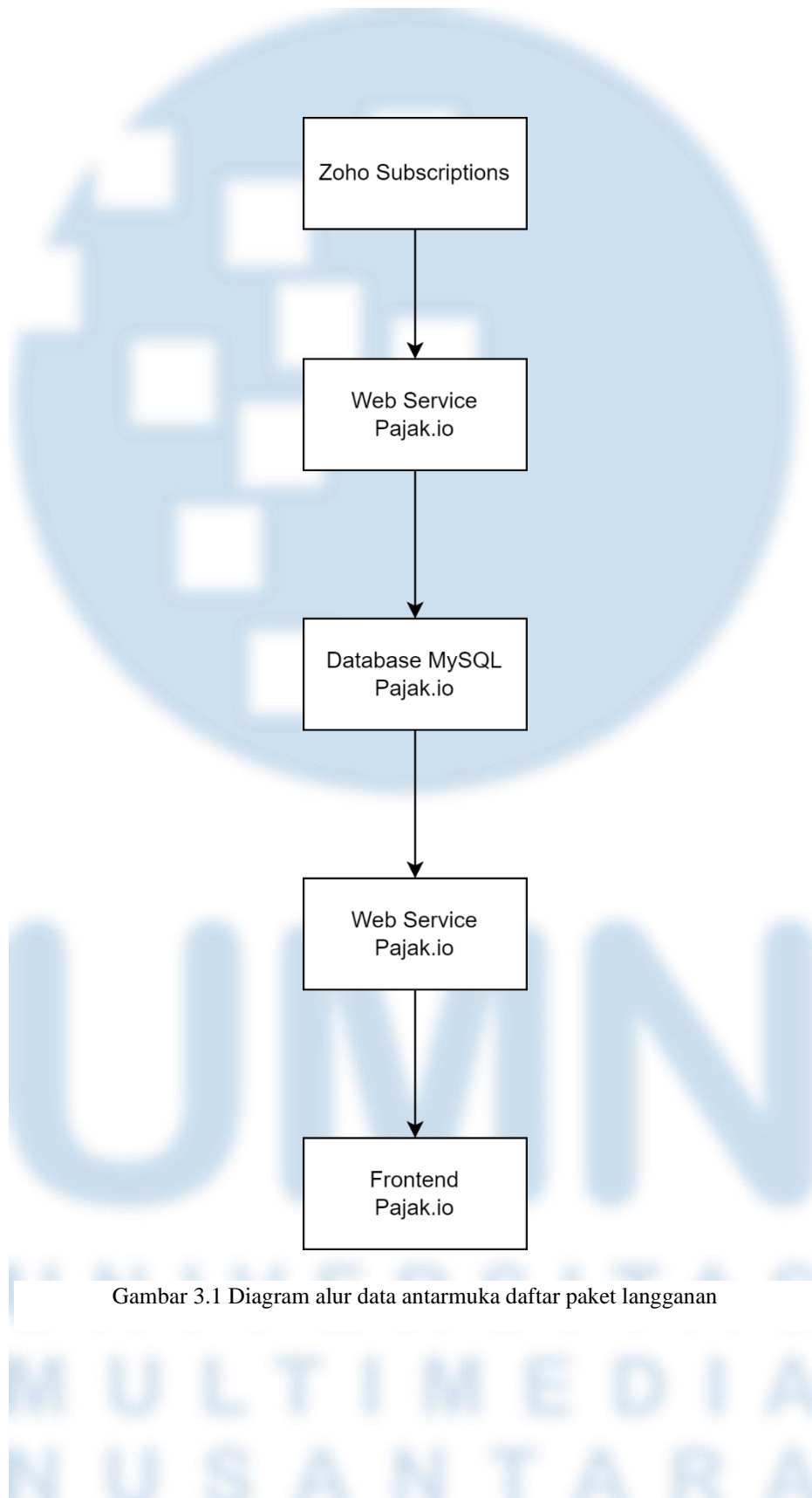
Selanjutnya, diperkenalkan dengan supervisor magang yang sekaligus menjadi ketua tim. Supervisor kemudian memerintahkan untuk mempelajari framework *Vue.js*, yaitu framework yang digunakan Pajak.io untuk mengimplementasikan antarmuka aplikasi *web*. Setelah itu, mempelajari standar *software engineering* yang dipakai di Pajak.io dengan membaca dan memahami dokumentasi yang sudah dibuat di Pajak.io agar mengerti terhadap standar praktisi yang dipakai di Pajak.io dan siap untuk melakukan pengembangan aplikasi *web*.

### **3.2.2.2 Mengimplementasikan antarmuka daftar paket langganan (Minggu 2-7)**

Kegiatan ini mengimplementasikan antarmuka daftar paket langganan dari bentuk *prototype* ke aplikasi *web*. Antarmuka daftar paket langganan adalah antarmuka yang berisi daftar layanan - layanan Pajak.io secara rinci beserta harga - harganya. Data - data pada antarmuka ini diambil dari Zoho Subscription, melalui *web service* Pajak.io. Setelah data diambil dari Zoho Subscription, data tersebut disimpan pada *database* MySQL dan diberikan kepada

*frontend* Pajak.io melalui *web service* Pajak.io lagi. *Web Service* ini dibuat oleh *Backend Engineer*. Sedangkan, yang mengurus data pada Zoho Subscriptions adalah tim *marketing* dan *sales*. Data akan disinkronisasi dari Zoho Subscriptions ke *database* Pajak.io secara berkala sehingga ketika tim *marketing* dan *sales* melakukan pembaruan data langganan, maka data di *database* Pajak.io juga terbaru, yang akhirnya juga membarui tampilan *website* Pajak.io secara otomatis. Gambar 3.1 merupakan diagram yang menjelaskan alur data dari antarmuka ini.



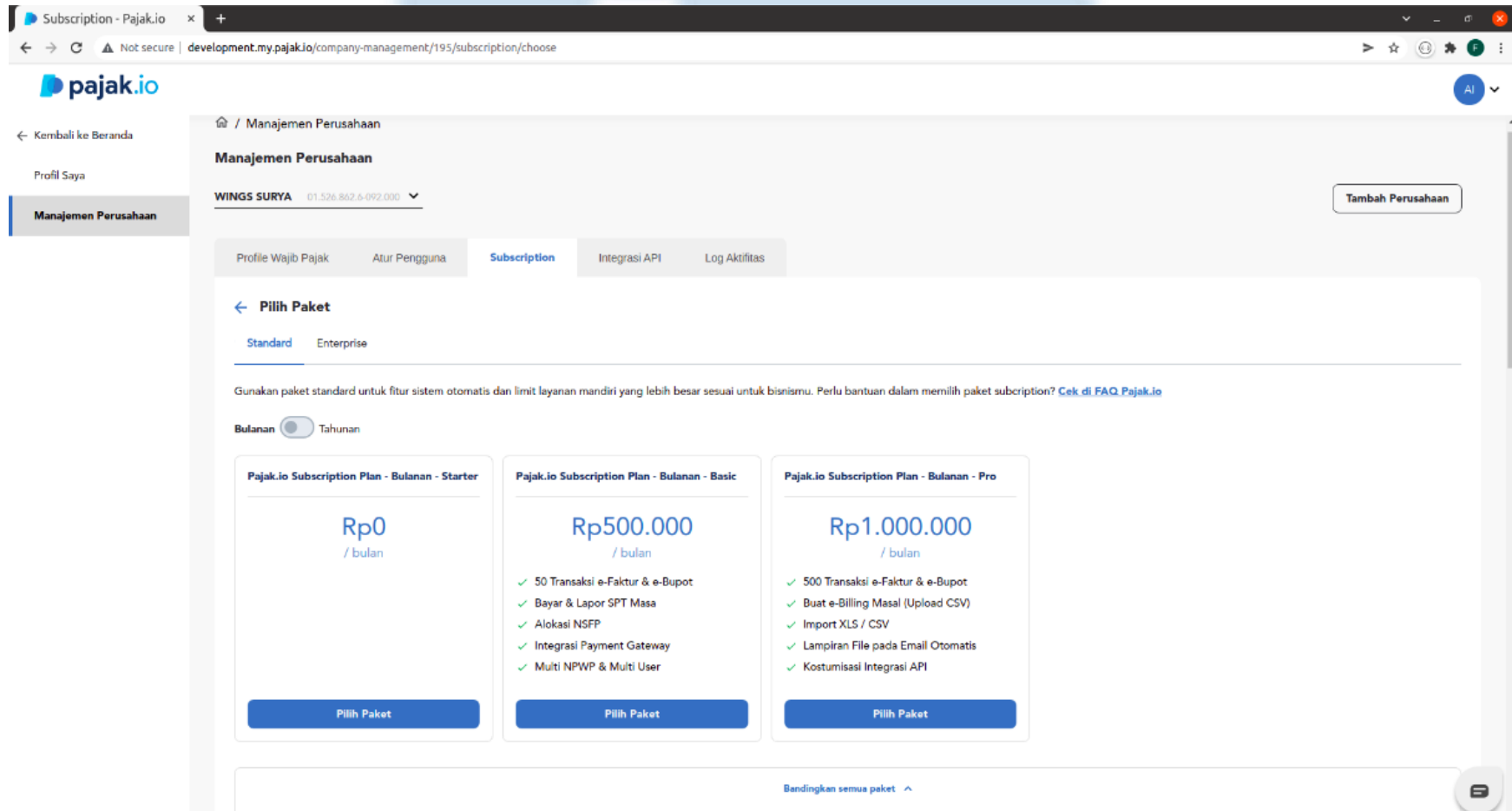


Gambar 3.1 Diagram alur data antarmuka daftar paket langganan

Antarmuka dibuat berdasarkan referensi *prototype* Figma dari tim *Product Design*. Pengembangan antarmuka dibuat dengan bahasa HTML, JavaScript, SCSS (*Syntactically Awesome Style Sheets*) dan *web framework* Vue.js. Proses *coding* juga bekerja sama dengan *Backend Engineer* dalam perancangan *web service* yang cocok digunakan dengan antarmuka *web* yang akan dibuat.

Setelah antarmuka selesai dibuat, kode yang ditulis diunggah ke *code repository* GitLab untuk dicek dan disetujui oleh *Engineering Lead*. Setelah disetujui, antarmuka *web* diberikan kepada *Product Manager* untuk dites oleh QA (*Quality Assurance Engineer*). Gambar 3.2 dan 3.3 merupakan hasil dari antarmuka *web* yang dibuat.





Gambar 3.2 Antarmuka daftar paket

Subscription - Pajak.io × +  
 development.my.pajak.io/company-management/195/subscription/choose

← Kembali ke Beranda  
 Profil Saya  
**Manajemen Perusahaan**

Bandingkan semua paket ^

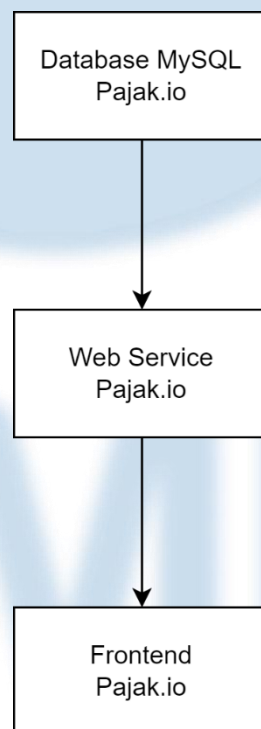
|                                    | Pajak.io Subscription Plan - Bulanan - Basic | Pajak.io Subscription Plan - Bulanan - Pro |
|------------------------------------|--|--|
| <b>e-Billing</b>                   |  |  |
| Buat e-Billing Massal (Upload CSV) | —  | ✓  |
| Bayar Pajak Langsung               | ✓  | ✓  |
| <b>e-Filing</b>                    |  |  |
| SPT Tahunan Badan                  | ✓  | ✓  |
| <b>e-Faktur</b>                    |  |  |
| Jumlah Faktur                      | 50   | 500  |
| Alokasi NSFP                       | ✓  | ✓  |
| Import Barang Jasa                 | —  | ✓  |
| Import Faktur Dengan QR Code       | —  | ✓  |
| Migrasi Data Faktur Historis       | —  | —  |
| <b>e-Bupot</b>                     |  |  |
| Jumlah Bupot                       | 500  | 50   |
| Migrasi Data Bupot Historis        | ✓  | —  |
| <b>Fitur Lainnya</b>               |  |  |
| Bayar SPT Masa                     | ✓  | ✓  |
| Lapor SPT Masa                     | ✓  | ✓  |
| Validasi NPWP Lawan Transaksi      | ✓  | —  |

Gambar 3.3 Antarmuka rincian paket



### 3.2.2.3 Mengimplementasikan antarmuka pembayaran (Minggu 8-13)

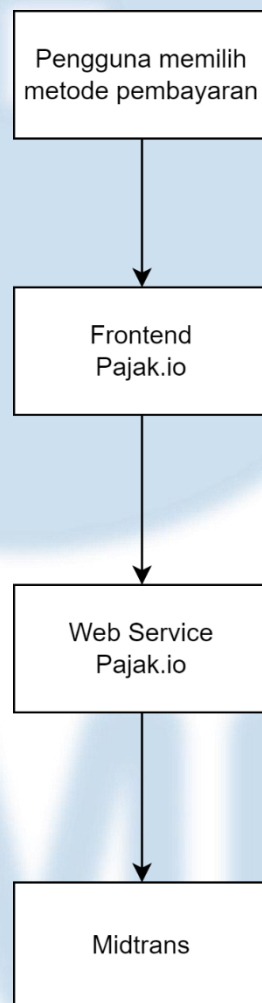
Kegiatan ini mengimplementasikan antarmuka pembayaran dari bentuk *prototype* ke aplikasi *web*. Antarmuka pembayaran adalah antarmuka yang berisi jumlah uang yang harus dibayar pengguna berdasarkan paket yang dipilih dan daftar metode pembayaran yang diterima oleh Pajak.io. Data - data pada antarmuka ini diambil dari *database* MySQL Pajak.io melalui *web service* Pajak.io yang dibuat oleh *Backend Engineer*. Gambar 3.2 merupakan diagram yang menjelaskan alur data dari antarmuka ini.



Gambar 3.2 Diagram alur data antarmuka pembayaran

Pada antarmuka ini, pengguna bisa memilih metode pembayaran yang akan digunakan. Setelah pengguna memilih, maka *frontend* akan membuat *HTTP request* ke *web service* Pajak.io untuk

membuat permintaan pembayaran dari Midtrans. Midtrans adalah layanan *payment gateway* untuk memproses pembayaran dari berbagai macam metode pembayaran yang digunakan oleh Pajak.io. Gambar 3.3 merupakan alur data setelah pengguna memilih metode pembayaran.



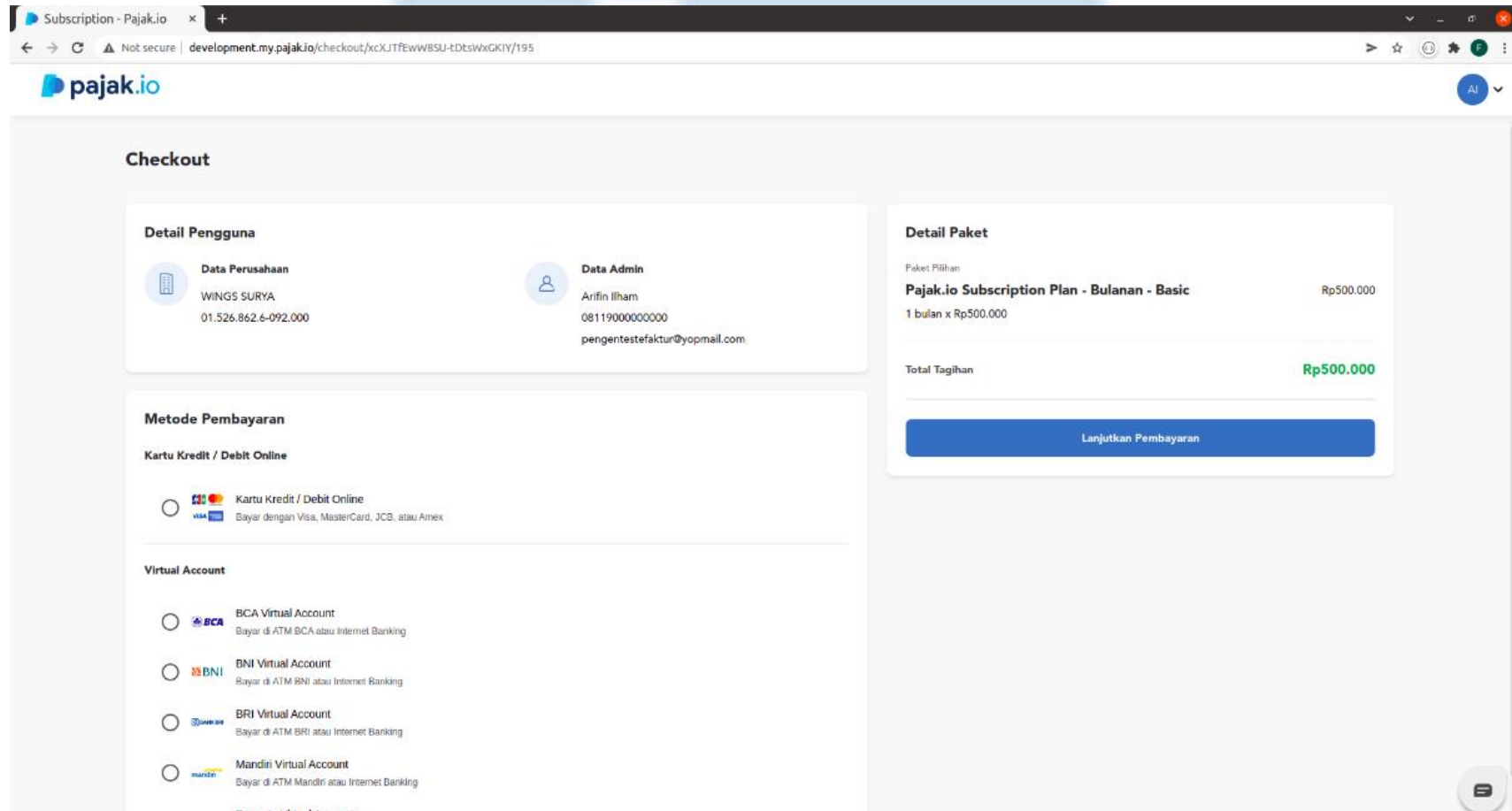
Gambar 3.3 Diagram alur data setelah pengguna memilih metode pembayaran

Antarmuka dibuat berdasarkan referensi *prototype* Figma dari tim *Product Design*. Pengembangan antarmuka dibuat dengan bahasa HTML, JavaScript, SCSS (*Syntactically Awesome Style Sheets*) dan *web framework* Vue.js. Proses *coding* juga bekerja

sama dengan *Backend Engineer* dalam perancangan *web service* yang cocok digunakan dengan antarmuka *web* yang akan dibuat.

Setelah antarmuka selesai dibuat, kode yang ditulis diunggah ke *code repository* GitLab untuk dicek dan disetujui oleh *Engineering Lead*. Setelah disetujui, antarmuka *web* diberikan kepada *Product Manager* untuk dites oleh *QA Engineer*. Gambar 3.4 merupakan hasil dari antarmuka *web* yang dibuat.

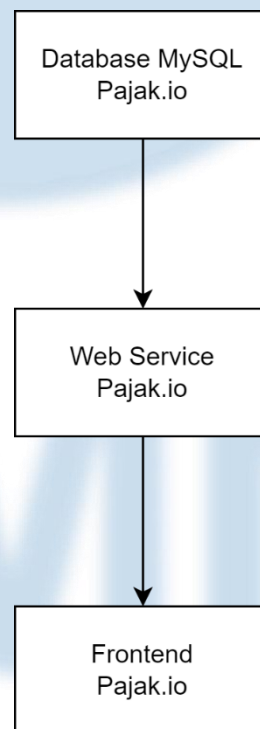




Gambar 3.4 Antarmuka pembayaran

### 3.2.2.4 Mengimplementasikan antarmuka status pembayaran (Minggu 14-20)

Kegiatan ini mengimplementasikan antarmuka pembayaran dari bentuk *prototype* ke aplikasi *web*. Antarmuka status pembayaran adalah antarmuka yang menampilkan jumlah yang harus dibayar beserta langkah - langkah untuk membayarnya dan daftar status pembayaran yang telah diminta. Setelah Data - data pada antarmuka ini diambil dari *database* MySQL Pajak.io melalui *web service* Pajak.io yang dibuat oleh *Backend Engineer*. Gambar 3.5 merupakan diagram yang menjelaskan alur data dari antarmuka ini.

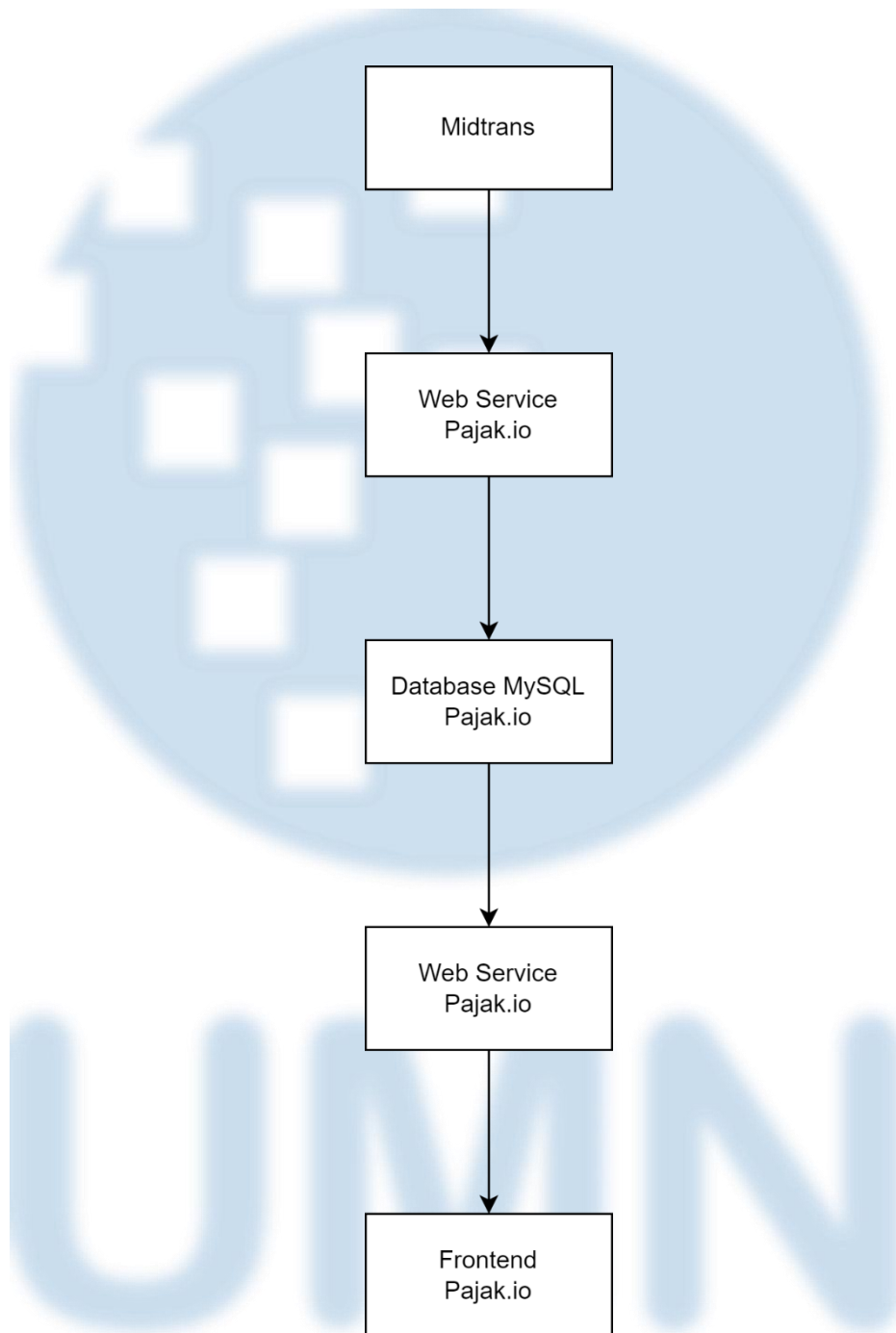


Gambar 3.5 Diagram alur data antarmuka status pembayaran

Setelah pengguna melakukan pembayaran, Midtrans, sebagai *payment gateway* akan memberitahu *web service* Pajak.io melalui *HTTP request* bahwa pembayaran sudah lunas beserta data

pengguna yang melakukan pembayaran tersebut. Setelah *web service* Pajak.io menerima data ini, maka *database* Pajak.io akan diperbarui dengan mengganti status pembayaran ke sukses. Hal ini juga akan terjadi ketika pengguna tidak membayar sampai batas waktu yang ditentukan. Ketika pengguna mengklik tombol “Cek Status Pembayaran” maka akan muncul pesan sukses. Namun, jika pengguna belum melakukan pembayaran, maka akan muncul pesan gagal. Gambar 3.6 merupakan diagram yang menjelaskan alur data setelah pembayaran sukses atau gagal.





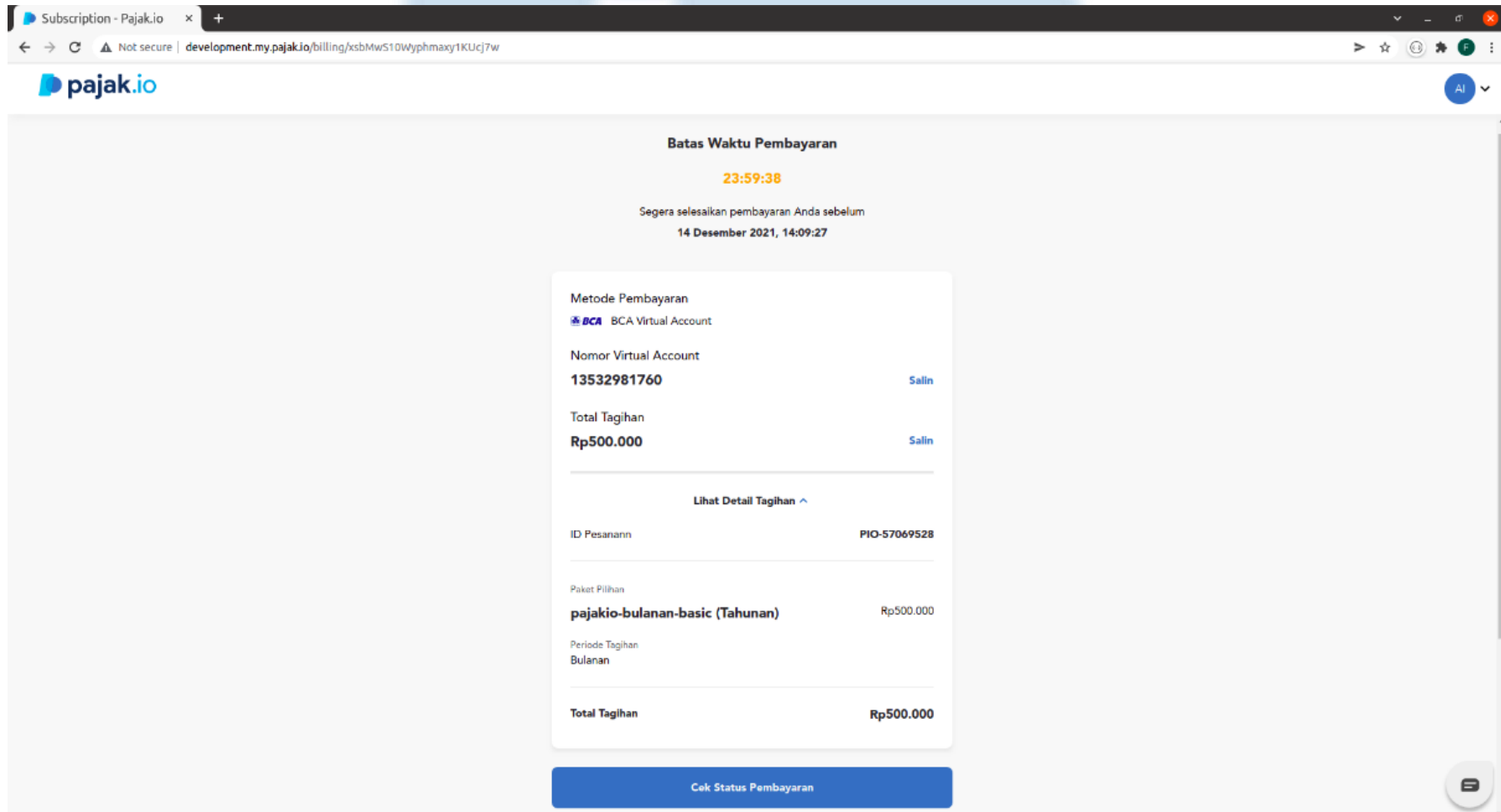
Gambar 3.6 Diagram alur data setelah pembayaran sukses atau gagal

Antarmuka dibuat berdasarkan referensi *prototype* Figma dari tim *Product Design*. Pengembangan antarmuka dibuat dengan bahasa HTML, JavaScript, SCSS (*Syntactically Awesome Style Sheets*) dan *web framework* Vue.js. Proses *coding* juga bekerja sama dengan *Backend Engineer* dalam perancangan *web service* yang cocok digunakan dengan antarmuka *web* yang akan dibuat.

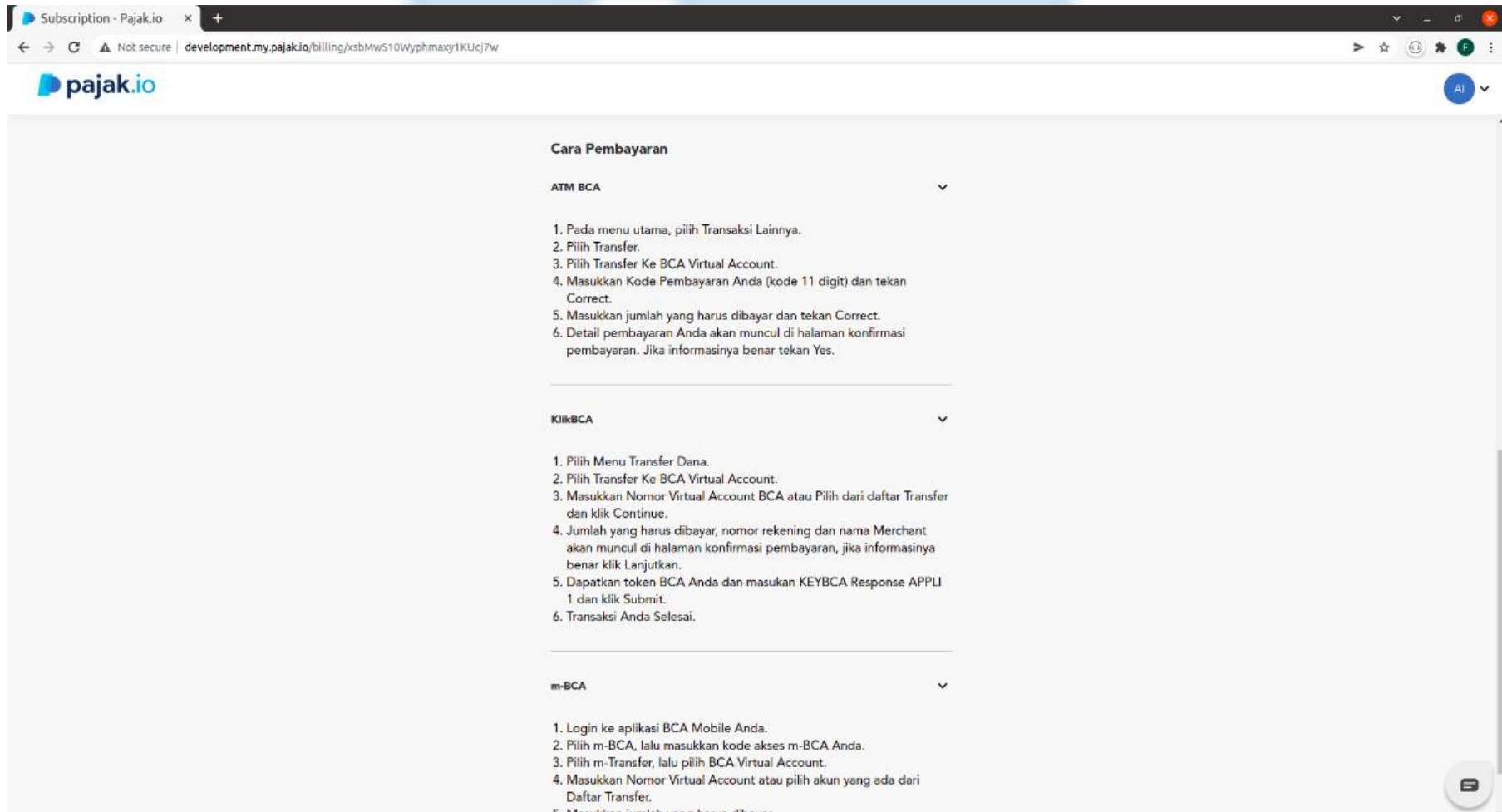
Setelah antarmuka selesai dibuat, kode yang ditulis diunggah ke *code repository* GitLab untuk dicek dan disetujui oleh *Engineering Lead*. Setelah disetujui, antarmuka *web* diberikan kepada *Product Manager* untuk dites oleh *QA Engineer*. Gambar 3.7, 3.8, 3.9, 3.10, 3.11 merupakan hasil dari antarmuka *web* yang dibuat.



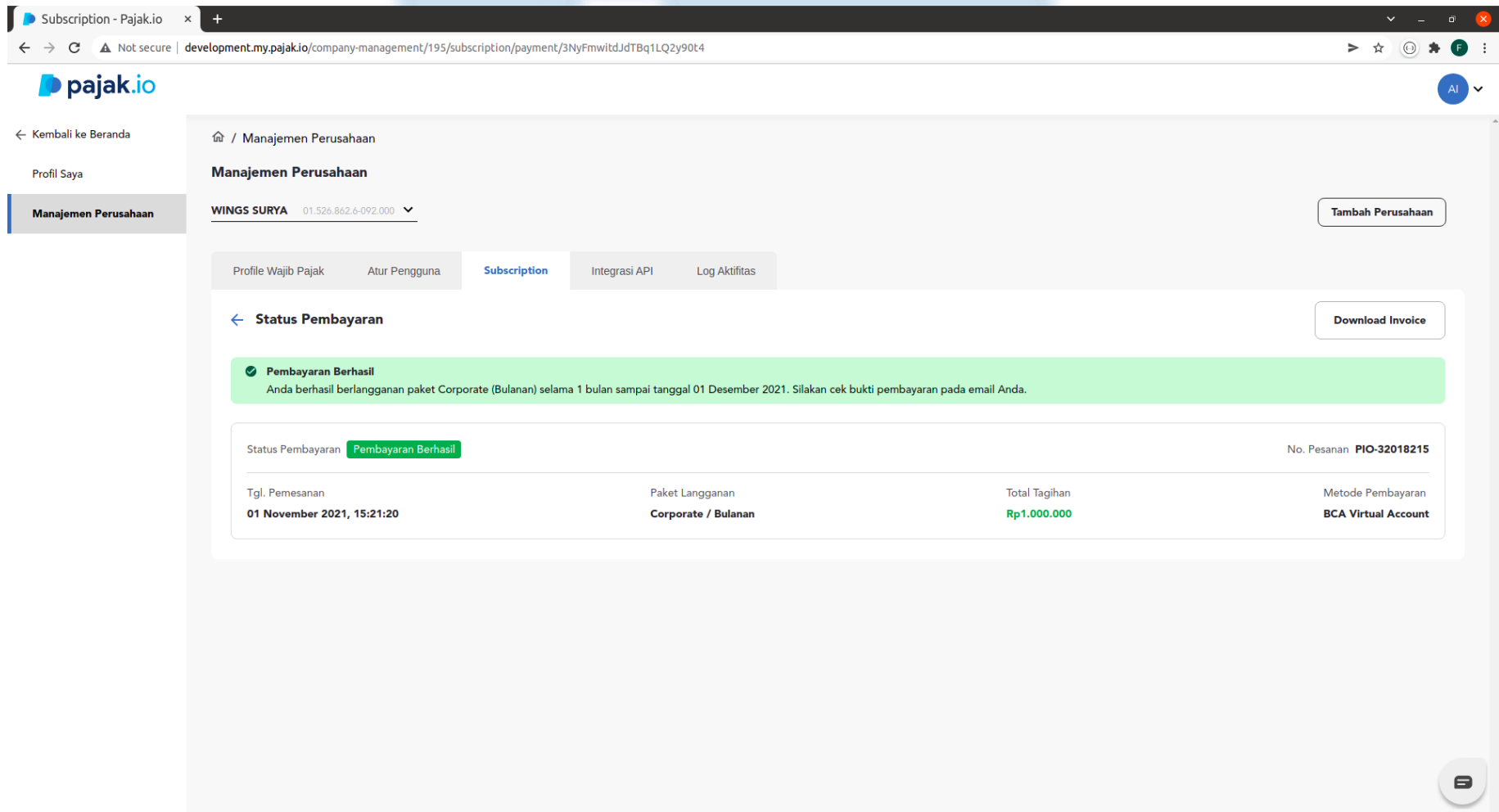




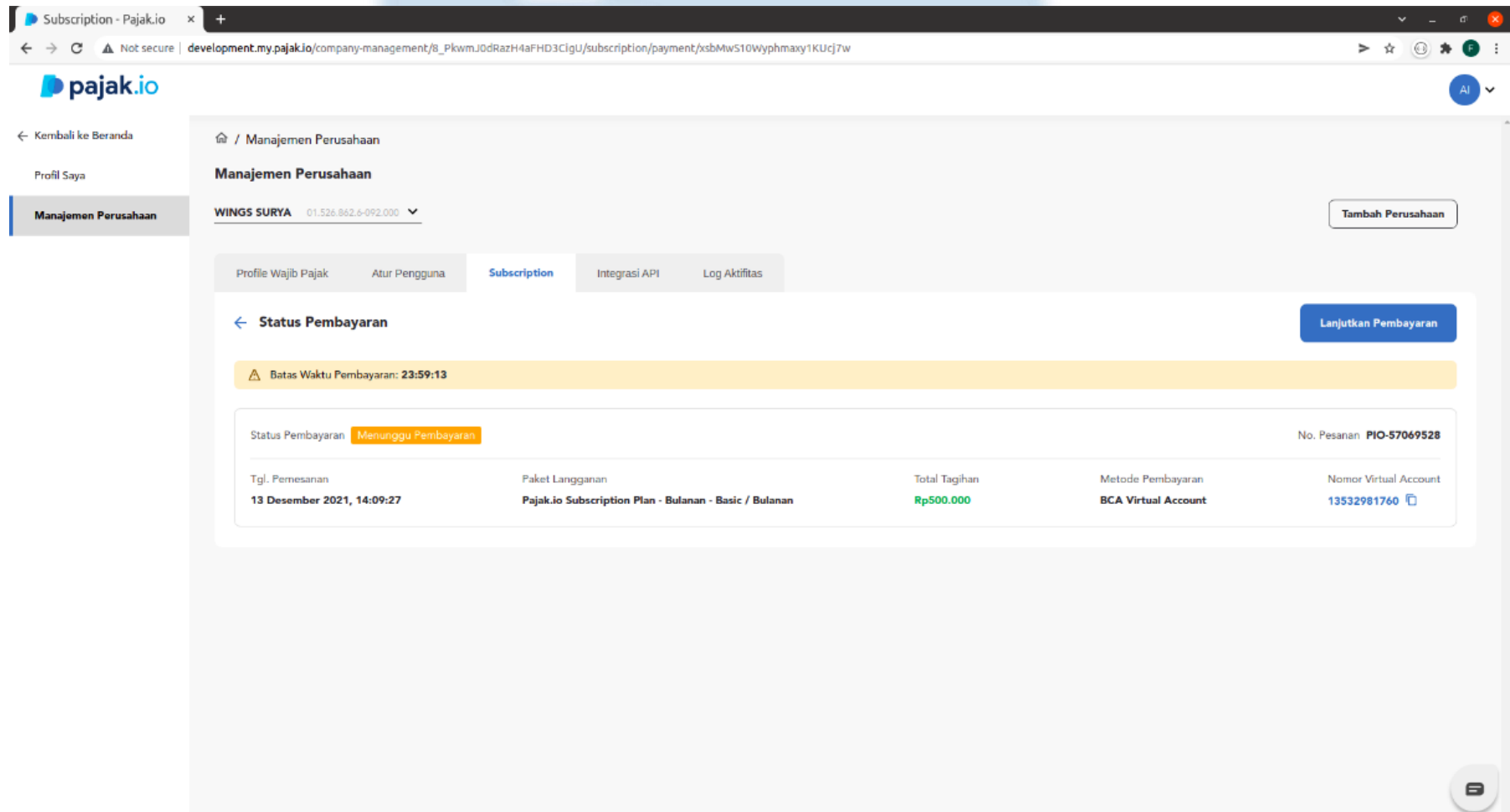
Gambar 3.7 Antarmuka status pembayaran



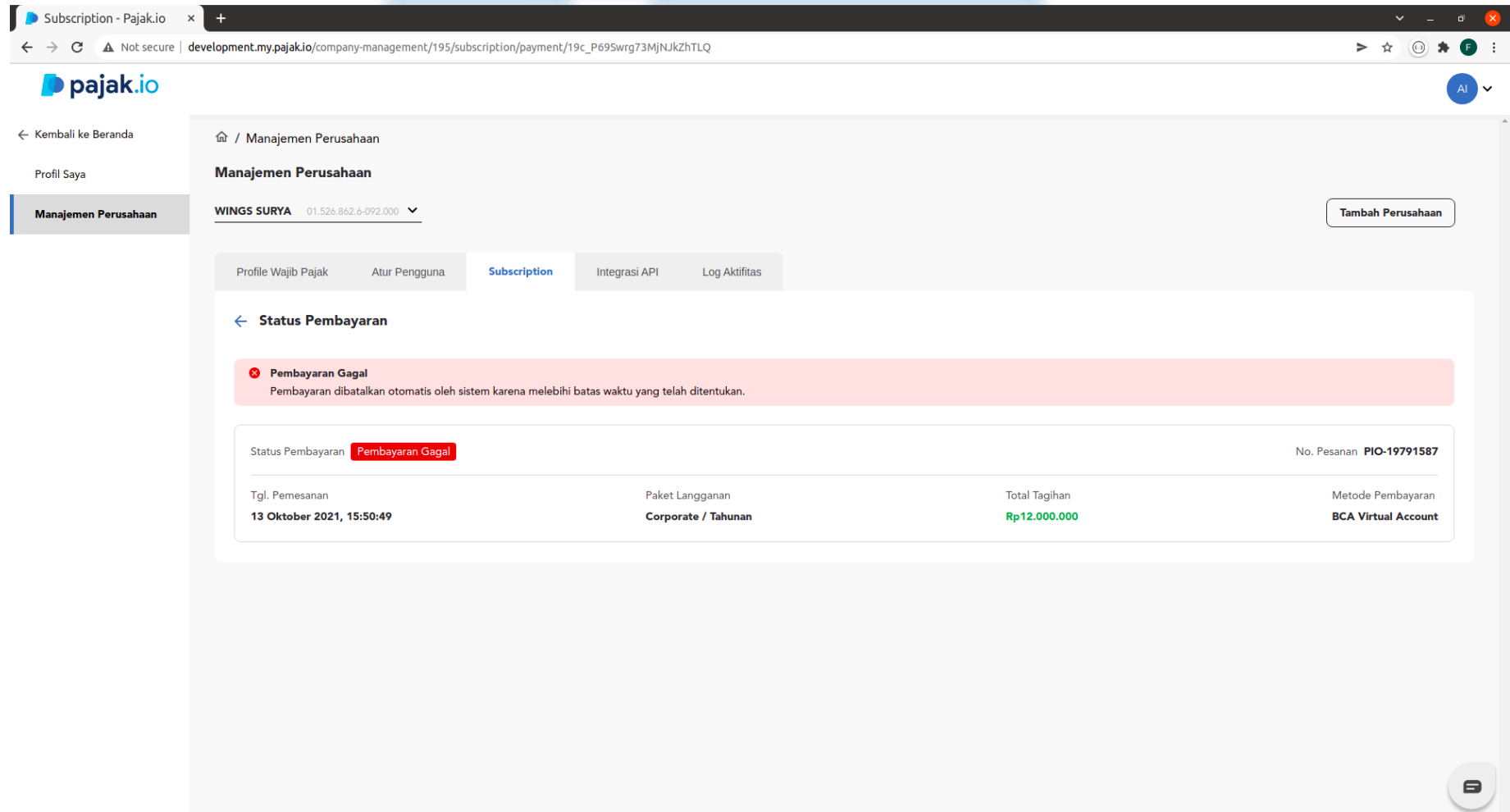
Gambar 3.8 Antarmuka cara pembayaran



Gambar 3.9 Antarmuka status pembayaran berhasil




Gambar 3.10 Antarmuka status pembayaran menunggu



Gambar 3.11 Antarmuka status pembayaran gagal

**Metode Pembayaran**

|   |                                   |                                   |  |
|---|-----------------------------------|-----------------------------------|--|
| Metode Pembayaran<br> <b>BCA</b> BCA Virtual Account | Harga Paket<br><b>Rp1.000.000</b> | Periode Tagihan<br><b>Bulanan</b> | Pembayaran Berikutnya<br><b>01 Desember 2021</b> |
|---|-----------------------------------|-----------------------------------|--|

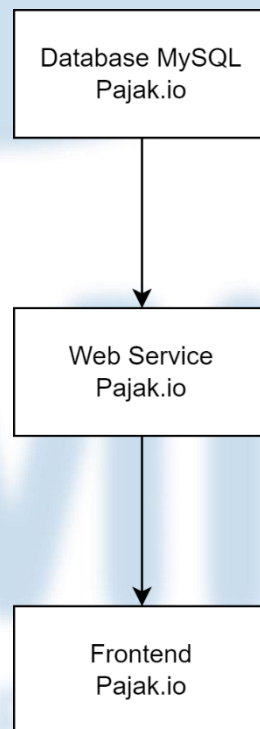
**Histori Pembayaran**

| No. Pesanan  | Tgl. Pemesanan    | Paket Langganan  | Metode Pembayaran   | Total Tagihan | Status              |          |                              |
|--------------|-------------------|--|---------------------|---------------|---------------------|----------|------------------------------|
| PIO-57069528 | 13/12/2021, 14:09 | Pajak.io Subscription Plan - Bulanan - Basic / Bulanan | BCA Virtual Account | Rp500.000     | Menunggu Pembayaran | 🕒 23h58m | <a href="#">Lihat Detail</a> |
| PIO-75352028 | 13/12/2021, 11:25 | Pajak.io Subscription Plan - Bulanan - Basic / Bulanan | BCA Virtual Account | Rp500.000     | Menunggu Pembayaran | 🕒 21h15m | <a href="#">Lihat Detail</a> |

Gambar 3.12 Antarmuka riwayat pembayaran

### 3.2.2.5 Mengimplementasikan antarmuka status penggunaan paket (Minggu 21-23)

Kegiatan ini mengimplementasikan antarmuka status penggunaan paket dari bentuk *prototype* ke aplikasi *web*. Antarmuka status penggunaan paket adalah antarmuka yang menampilkan penggunaan kuota paket Pajak.io yang dibeli dan digunakan. Selain itu, antarmuka ini juga menampilkan data riwayat pembayaran beserta metode pembayaran yang digunakan untuk membayar paket sebelumnya. Data - data pada antarmuka ini diambil dari *database* MySQL Pajak.io melalui *web service* Pajak.io yang dibuat oleh *Backend Engineer*. Gambar 3.13 merupakan diagram yang menjelaskan alur data dari antarmuka ini.



Gambar 3.13 Diagram alur data antarmuka status penggunaan paket

Antarmuka dibuat berdasarkan referensi *prototype* Figma dari tim *Product Design*. Pengembangan antarmuka dibuat dengan bahasa HTML, JavaScript, SCSS (*Syntactically Awesome Style Sheets*) dan *web framework* Vue.js. Proses *coding* juga bekerja sama dengan *Backend Engineer* dalam perancangan *web service* yang cocok digunakan dengan antarmuka *web* yang akan dibuat.

Setelah antarmuka selesai dibuat, kode yang ditulis diunggah ke *code repository* GitLab untuk dicek dan disetujui oleh *Engineering Lead*. Setelah disetujui, antarmuka *web* diberikan kepada *Product Manager* untuk dites oleh *QA Engineer*. Gambar 3.14 dan 3.15 merupakan hasil dari antarmuka *web* yang dibuat.





The screenshot displays the 'Subscription' page for a Corporate plan. The current subscription is 'Corporate' with a price of Rp1.000.000 and a monthly billing cycle. The next payment is due on 01 Desember 2021. The payment history table shows two transactions:

| No. Pesanan  | Tgl. Pemesanan    | Paket Langganan  | Metode Pembayaran   | Total Tagihan | Status                       |
|--------------|-------------------|--|---------------------|---------------|------------------------------|
| PIO-57069528 | 13/12/2021, 14:09 | Pajak.io Subscription Plan - Bulanan - Basic / Bulanan | BCA Virtual Account | Rp500.000     | Menunggu Pembayaran @ 23h58m |
| PIO-75352028 | 13/12/2021, 11:25 | Pajak.io Subscription Plan - Bulanan - Basic / Bulanan | BCA Virtual Account | Rp500.000     | Menunggu Pembayaran @ 21h15m |

Gambar 3.14 Antarmuka status penggunaan paket

**Metode Pembayaran**

Metode Pembayaran  
**BCA** BCA Virtual Account

Harga Paket  
**Rp1.000.000**

Periode Tagihan  
**Bulanan**

Pembayaran Berikutnya  
**01 Desember 2021**

**Histori Pembayaran**

| No. Pesanan  | Tgl. Pemesanan    | Paket Langganan  | Metode Pembayaran   | Total Tagihan | Status   |
|--------------|-------------------|--|---------------------|---------------|--|
| PIO-57069528 | 13/12/2021, 14:09 | Pajak.io Subscription Plan - Bulanan - Basic / Bulanan | BCA Virtual Account | Rp500.000     | Menunggu Pembayaran <span>🕒 23h58m</span> <a href="#">Lihat Detail</a> |
| PIO-75352028 | 13/12/2021, 11:25 | Pajak.io Subscription Plan - Bulanan - Basic / Bulanan | BCA Virtual Account | Rp500.000     | Menunggu Pembayaran <span>🕒 21h15m</span> <a href="#">Lihat Detail</a> |

Gambar 3.15 Antarmuka riwayat pembayaran

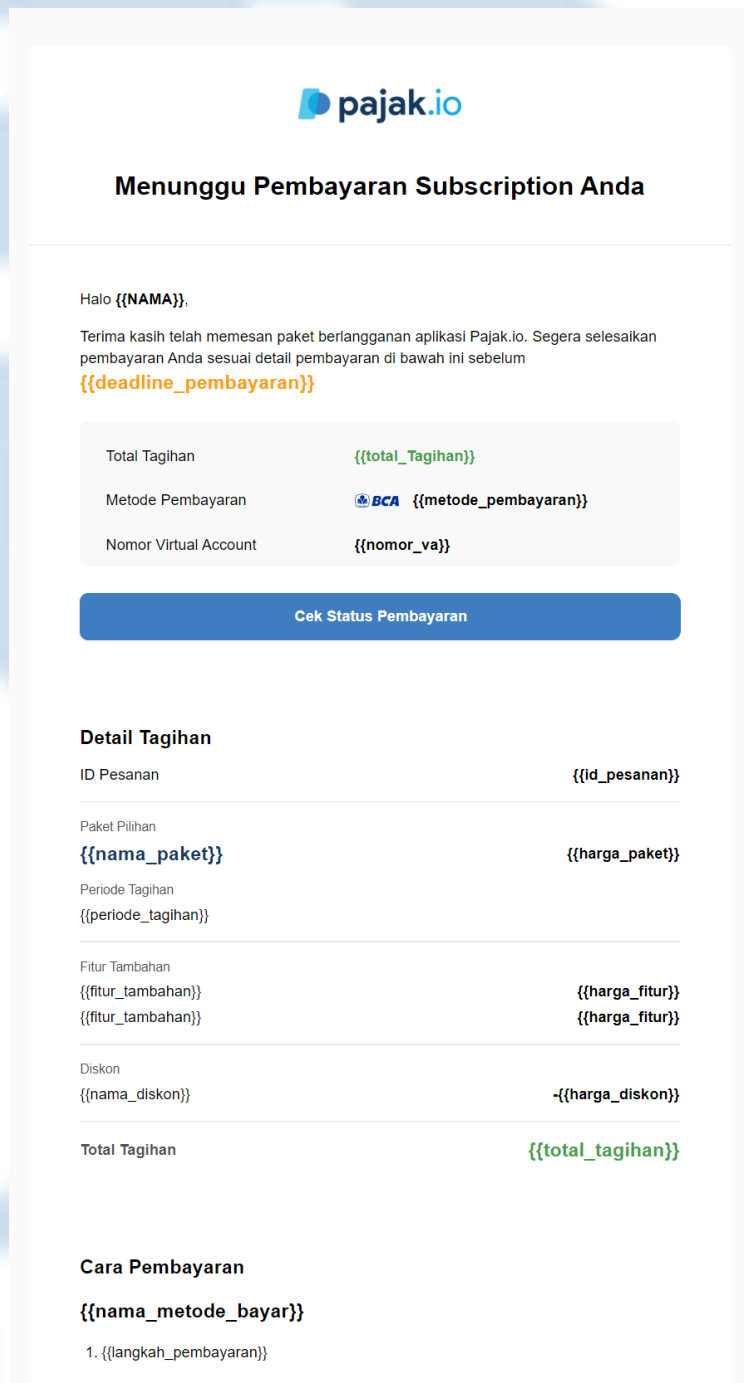
### 3.2.2.6 Mengimplementasikan email – email yang berkaitan dengan langganan (Minggu 24-26)

Kegiatan ini mengimplementasikan antarmuka email – email yang berkaitan dengan langganan dari bentuk *prototype* ke bentuk email yang bisa dikirim dari *backend website*. Email ini berguna sebagai notifikasi bagi pengguna Pajak.io ketika melakukan kegiatan yang berkaitan dengan langganan. Data - data pada email ini diambil dari *database* MySQL Pajak.io secara otomatis dari *backend* sesuai dengan penerima email. Berikut adalah daftar email yang dibuat:

- a) Email instruksi pembayaran
- b) Email pembayaran sukses
- c) Email langganan berakhir

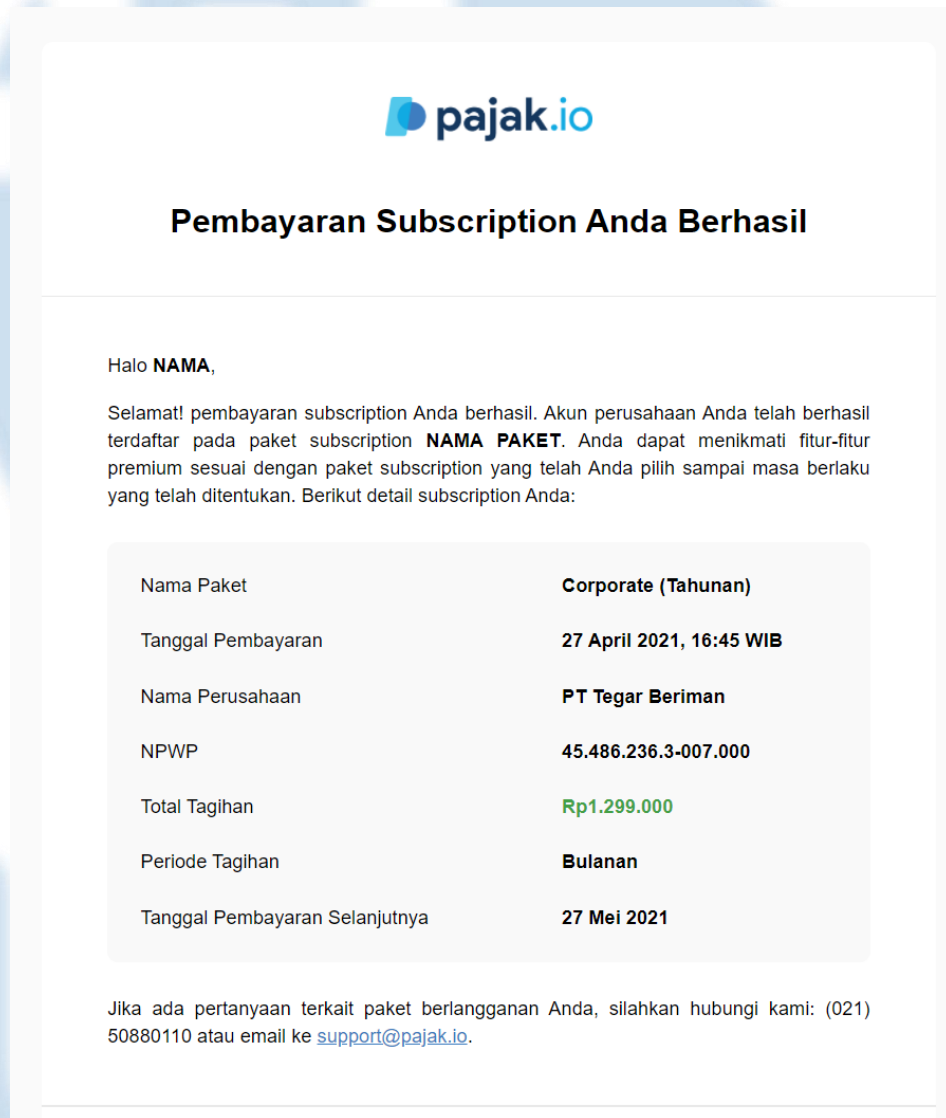
Antarmuka dibuat berdasarkan referensi *prototype* Figma dari tim *Product Design*. Pengembangan antarmuka dibuat dengan bahasa HTML dan CSS. Proses *coding* juga bekerja sama dengan *Backend Engineer* dalam perancangan *web service* yang cocok digunakan untuk mengirim email.

Gambar 3.16 adalah email instruksi pembayaran yang dikirim setelah pengguna melakukan pembelian paket.



Gambar 3.16 Email instruksi pembayaran

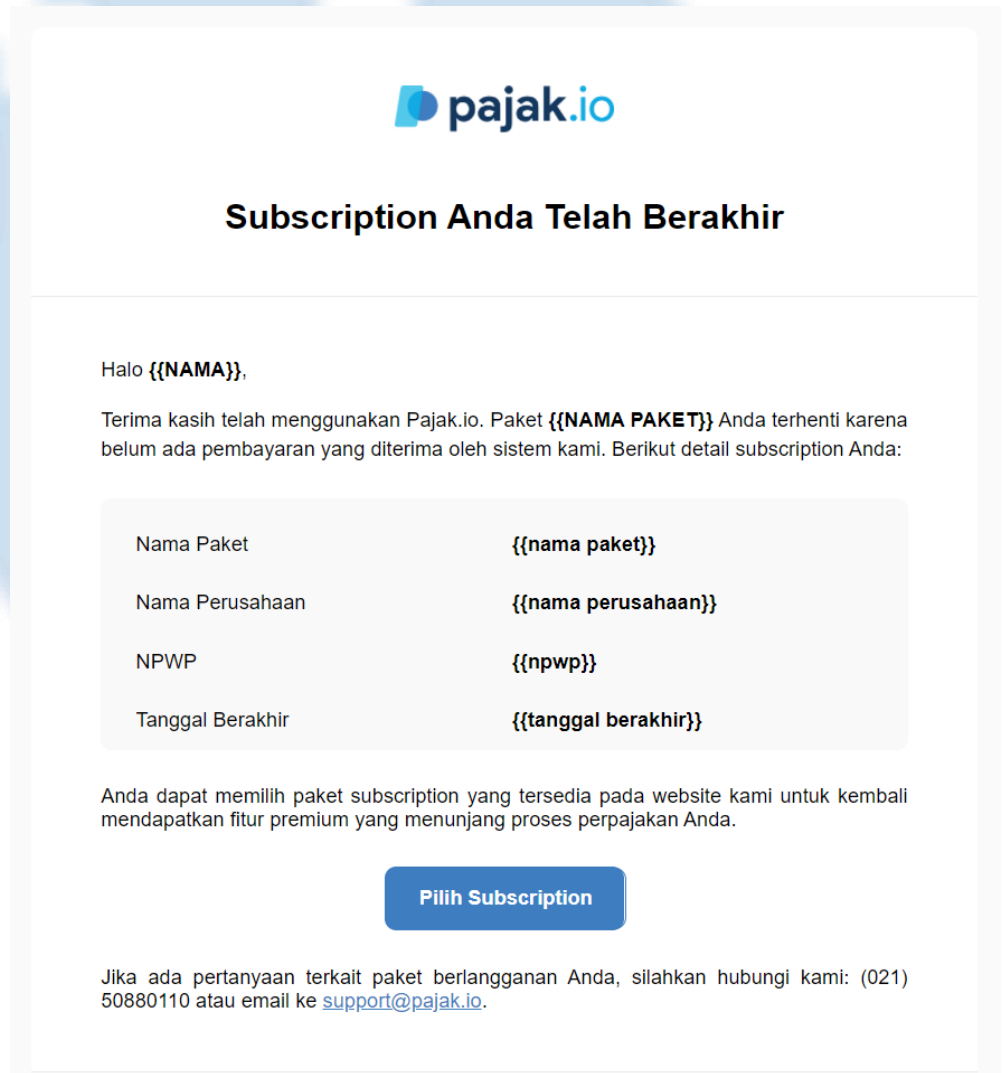
Gambar 3.13 adalah email pembayaran sukses yang dikirim setelah pengguna membayar tagihan.



Gambar 3.17 Email pembayaran sukses

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

Gambar 3.14 adalah email langganan berakhir yang dikirim setelah pengguna mengakhiri langganan Pajak.io.



Gambar 3.18 Email langganan berakhir



### 3.2.2.7 Presentasi akhir magang dan offboarding (Minggu 27)

Pada minggu akhir magang, Presentasi disiapkan sesuai dengan bentuk presentasi akhir magang Pajak.io yang berisi cerita pengalaman magang, saran, masukan, kritik terhadap perusahaan, produk Pajak.io, serta orang - orang didalamnya. Setelah itu, presentasi dilakukan kepada atasan - atasan yang berkaitan dengan posisi magang, dimulai dari CTO (*Chief Technology Officer*), CPO (*Chief Product Officer*), *Product Manager*, serta *Engineering Lead*. Setelah presentasi, *offboarding* dilakukan dengan menerima sertifikat magang dari perusahaan dan menyelesaikan administrasi – administrasi magang yang diperlukan oleh kampus dari perusahaan.

### 3.2.3 Kendala yang Ditemukan

- 1) Fitur – fitur yang dikerjakan oleh *Engineer* kurang diuji coba secara sempurna oleh QA karena *Product Manager* merangkap pekerjaan QA. Akibatnya, sering banyak kesalahan dan *bug* yang muncul belakangan dan bukan pada saat masa uji coba, yang akhirnya menambah pekerjaan di saat yang tidak tepat.
- 2) Koneksi VPN internal yang sering terputus. Masalah ini juga dialami beberapa karyawan lain di dalam Fintax. Akibatnya, para karyawan yang bekerja dari rumah tidak dapat mengakses *web service* internal dengan lancar, yang akhirnya mengganggu dan memperlambat pekerjaan.
- 3) Ketua tim terkena COVID-19 dan tidak ada gantinya. Dalam satu proyek, hanya ketua tim yang memegang hak akses *deploy* untuk mengupload aplikasi. Selain itu, ia juga yang memimpin dan mengorganisir jalannya *sprint*. Pada saat magang, ketua tim terkena penyakit COVID-19 dan tidak bisa bekerja selama satu bulan sehingga pekerjaan di dalam satu tim tersebut terhambat.

### 3.2.4 Solusi atas Kendala yang Ditemukan

- 1) *Product Manager* akhirnya memperkerjakan orang lain untuk melakukan uji coba. Sehingga proses uji coba untuk setiap fitur baru yang dikembangkan bisa lebih rinci dan fokus, yang akhirnya mempermudah proses perbaikan jika ada masalah.
- 2) Melaporkan masalah ini kepada tim infrastruktur. Lalu tim infrastruktur merespon dengan meningkatkan atau *upgrade* server VPN dan memberitahukan semua karyawan untuk meningkatkan klien VPNnya sehingga koneksinya bisa lebih stabil. Ketika VPN lebih stabil, maka pekerjaan pengembangan fitur menjadi tidak terganggu lagi.
- 3) Menerima hak akses *deploy* untuk menggantikan ketua tim yang sakit. Sehingga proses *deploy* bisa dilakukan tanpa ketua tim, dan proses pengembangan bisa berjalan dengan lancar. Hak akses ini terbagi menjadi dua, yaitu untuk *backend* dan *frontend*, masing-masing hak akses ini diberikan kepada posisi yang bersangkutan dan tidak kepada satu orang seperti ketua tim. Selain itu, yang memimpin dan mengorganisir jalannya rapat *sprint* diganti menjadi *Product Manager*. Perubahan ini dilakukan secara sementara sampai ketua tim dapat melakukan pekerjaannya kembali.